BioE143/243 Computational Methods in Biology
UNIX Tutorial

Part I

pwd (path display)

It shows the absolute path of your current directory.

whoami or **who am i** (current user display)

It shows the name of the current user.

ls (list files and directories in the current directory)

l – shows more information such as time, date, owner, group, authorization of files and directories, etc. in list format

t – sort the files and directories by time

a – show hidden files and directories

Examples:

To show all hidden files and directories in list format and sort them by time: **ls –lat**

TO FIND all files that match a certain pattern, use *

ls *.cpp

will list all files that end in .cpp

ls ran*

will list all files that start with ran

ls ran*.cpp

will list all files that start with ran and end with .cpp, with anything in between.

cp (copy file(s) or directory(ies))

r – recursive (use it when you want to copy a directory from one directory to another)

Examples:

To copy a file from the current directory to another:

cp myfile mydir/

To copy a directory from the current directory to another:

cp –r mydir nextdir/

To copy a file from somewhere else to the current directory:

cp /mydir1/mydir2/myfile .

(the period (.) tells the system to place the file in the current directory)

mv (move a file or a directory to another location or rename them)

Examples:

To move a file from the current directory to another:

mv myfile mydir/

To move a directory from the current directory to another:

mv mydir nextdir/

(you do not need recursive (**r**) flag to move a directory. Actually, **mv** command does not have **r** flag.)

To rename a file: **mv myfile mynewfile**

To rename a directory: **mv mydir mynewdir**

mkdir (make diretory)

To make *mydir* directory: **mkdir mydir**

cd (change directory)

Examples:

To change to *mydir* directory: **ch mydirectory**

To move up one directory level: **ch ..**

~ (take me to my home directory)

This is NOT a command. It let you go back to your home directory in one command. The following examples will show how it is used.

Examples:

To change to my home directory: **cd ~**

To copy a file from the current directory to my home directory:

cp myfile ~/

rm (remove file(s) or directory(ies))

r – recursive (use it when you want to remove directory(ies))

l – confirm with user before permenantly remove selected files or/and directory(ies)

Examples:

To remove a file: **rm myfile**

To remove a file with confirmation: **rm –l myfile**

To remove a directory with confirmation: **rm –rl mydir**

To remove all files in the current directory: **rm ***

To remove all files and all directories in the current directory: **rm –r ***

Note: The default of **rm** command on all the DECF computers has been set to **–l**. To skip the confirmation of deletion, put **** before **rm** command. For example, **\rm myfile**. Be EXTREMELY careful when you use **rm** command. There is no turning back after the file(s) or directory(ies) are removed.

tab key (complete the name of a file or a directory)

If you have a filename called *myfile*, and at the command line you type *myf* then hit tab key, the system will complete *myfile* at the command line.

emacs (text editor program)

Example:

To create a new file: emacs myfile

To edit an existing file: emacs myfile (same as above)

To save a file inside emacs: CTRL-x s y

To exit emacs: CTRL-x-c

tar (archive files)

c – to create a tar file

x – extract the contents of a tar file

f – specifies the filename

z – use zip/gzip to compress the tar file or to read from a compressed tar file

v – verbose; show files being tarred or extracted

Examples: (NO dash before flags)

To create a tar file: **tar cvf goody.tar *.txt**

(tar all files having *.txt* as the last 4 characters in the filename)

To create a compressed tar file: **tar cvzf goody.tzg *.txt**

(same as the above, plus compressing the tar file)

To extract a tar file: **tar xvf goody.tar**

To extract a compressed tar file: **tar xvzf goody.tzg**

(you cannot extract a compressed tar file without **z** flag)

Note: If you use **f** flag when creating a tar file, but forgot to specify a filename, tar program will use the filename of the first file being

tarred as the tar file's filename and replace the original file. So BE CAREFUL!!

gzip [aka **gunzip**] (to compress/decompress a file)

f – to force to create a gz file

d – to decompress a gz file

Examples:

To compress a file: **gzip –f goody.tar**

(gz program's default is to keep the original filename and add *.gz* at the end. So the gz file here will be *goody.tar.gz*)

To decompress a gz file: **gzip –d goody.tar.gz**

(the decompressed file will be named as *goody.tar* here)

Note: If you have created a ZIP file in a Windows system and want to unzip the ZIP file in UNIX, use **unzip** command. The default is set to unzip a ZIP file, so you do not need to use any flags. For example, **unzip windowsxp.zip**

Part II

[$ sign means command prompt]

- Source code → [compiler] → binary code (executables)
- File extension of C code: .c
- File extension of C++ code: .cpp
- File extension of Java code: .java
- Command to call C compiler: gcc
- Command to call C++ compiler: g++
  - ($ g++ mycode.cpp –o mycode.exe)
  - ($ ./mycode)
- Command to call java compiler: javac
  - ($ javac myapp.java)
  - ($ java myapp)
- To run program in background:
  - ($ ./mycode > message.log &)
- Take arguments at command prompt:
  - ($ ./mycode arg1 arg2)
- Combine last two points:

- ($ ./mycode arg1 arg2 > message.log &)

REMOTE ACCESS/FILE TRANSFER

To connect remotely to another machine use ssh

- $ ssh be243@kepler.berkeley.edu

You will notice that using text editors (emacs) remotely does not allow the use of graphics, i.e. no drop down menus, so you have to know how to move around in the window using emacs commands (see emacs tutorial, or if you are in emacs, type control-h t to enter its tutorial)
In order to use the graphical interface remotely, use the X-terminal on macs, and on windows machines you need to download an extra program such as exceed.

To transfer files across machines use either sftp

- $ sftp be243@kepler.berkeley.edu

  Move to whichever directory you want to put or get your file

- > cd hw2

  Use the put or get command.

- > put myfile
- > get anotherfile
- > exit (or quit)

  To speed this up, you can use the * function, described in the ls section at the top.

- >put *.cpp

Also you can use scp command to move files onto another computer

- scp myfile be243@kepler.berkeley.edu:hw2/

Both these methods are secure and will request your password.

RUNNING JOBS

Once you have submitted a program, and are say running it in the background, You can check if it is still running by using

- top

  which will tell what % cpu is using for each job. Type Control-c to exit

- ps

a shorter list of what is currently running

If you have question about using any commands you can use the manual

- man ls

This will tell you what the command ls does, and what flags you can use with it. It refers to the manual pages and should be available for all commands